



**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS**  
**DEPARTAMENTO DE COMPUTAÇÃO**  
**LINGUAGEM DE PROGRAMAÇÃO**  
**4º Bimestre - Atividade Prática 1**

**MANIPULAÇÃO DE ARQUIVOS DE TEXTO**

Em Java cada arquivo é um fluxo sequencial de bytes;

O sistema operacional é responsável por indicar ao programa em Java quando o arquivo alcança o final do fluxo (o programa não precisa saber como S.O. faz isso);  
Todo arquivo deve ser aberto (*open*) antes de ser lido/escrito e fechado (*close*) após a leitura/escrita;

Um programa Java abre o arquivo associando um objeto ao local onde se encontra o “fluxo de bytes”;

O processamento de texto é feito pelas classes no pacote java.io:

- **FileInputStream** e **FileOutputStream** (entrada/saída baseada em bytes)
- **FileReader** e **FileWriter** (entrada/saída baseada em caracteres)
- **BufferedReader** e **BufferedWriter** (entrada/saída de texto a partir de um fluxo de caracteres)

Compile, analise e teste o problema abaixo:

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class ManipuladorArquivo {
    private final String caminhoArquivo;

    public ManipuladorArquivo(String caminhoArquivo) {
        this.caminhoArquivo = caminhoArquivo;
    }

    public String lerTudo() throws IOException, FileNotFoundException {
        FileReader fileReader = new FileReader(caminhoArquivo);
        BufferedReader readerf = new BufferedReader(fileReader);
        String linha = readerf.readLine();
        StringBuilder construtorString = new StringBuilder();
        while (linha != null) {
            construtorString.append(linha);
            construtorString.append("\n");
            linha = readerf.readLine();
        }
        readerf.close();

        return construtorString.toString();
    }

    public void escrever(String texto, boolean apagarConteudoArquivo) throws IOException {
        FileWriter fw = new FileWriter(caminhoArquivo, !apagarConteudoArquivo);
        BufferedWriter fwriter = new BufferedWriter(fw);
        fwriter.write(texto + "\n");
        fwriter.close();
    }

    public void escrever(String texto) throws IOException {
        escrever(texto, false);
    }

    public static void main(String[] args) {
        try {
            String caminho = "C:\\\\LLP\\\\teste.txt"; // altere o diretório onde o arquivo será
            salvo
            ManipuladorArquivo manipuladorArquivo = new ManipuladorArquivo(caminho);
            Scanner scanner = new Scanner(System.in);
            System.out.println("Digite uma informação para ser inserida ao arquivo: ");
            manipuladorArquivo.escrever(scanner.nextLine());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println("\nConteúdo lido do arquivo: \n" +
manipuladorArquivo.lerTudo());
    } catch (IOException ex) {
        System.out.println("Erro: " + ex.getMessage());
    }
}
```

Utilizando a classe ManipuladorArquivo, faça um programa que leia o caminho (diretório) e o nome de um arquivo de texto. Além disso, seu programa deve ter um menu para o usuário:

- Ver na tela todo o conteúdo do arquivo;
- Adicionar uma nova linha ao final do arquivo;
- Sair do programa.